A tutorial on how to compute traditional IAT effects with R 📊 ⬡

Jessica Röhner [a,✉] and Philipp J. Thoss [b]

[a]University of Bamberg, Germany
[b]Chemnitz, Germany

Abstract ■ The Implicit Association Test (IAT) is the most frequently used and the most popular measure for assessing implicit associations across a large variety of psychological constructs. Altogether, 10 algorithms have been suggested by the founders of the IAT to compute what can be called the traditional IAT effects (i.e., the six D measures: D1, D2, D3, D4, D5, D6, and the four conventional measures [C measures]: C1, C2, C3, C4). Researchers can decide which IAT effect they want to use, whereby the use of D measures is recommended on the basis of their properties. In this tutorial, we explain the background of the 10 traditional IAT effects and their mathematical details. We also present R code as well as example data so that readers can easily compute all of the traditional IAT effects. Last but not least, we present example outputs to illustrate what the results might look like.

Keywords ■ Implicit Association Test, IAT, traditional IAT effects, D measures, Conventional measures, C measures. **Tools** ■ R.

✉ jessica.roehner.perspsych@uni-bamberg.de

JR: 0000-0003-0633-3386; *PJT*: 0000-0002-4672-2792

Introduction

The Implicit Association Test (IAT Greenwald, McGhee, & Schwartz, 1998) is a very popular and frequently used indirect measure (e.g., Bosson, Swann, & Pennebaker, 2000; Rudolph, Schröder-Abé, Schütz, Gregg, & Sedikides, 2008) that has quickly established its validity as a general method for assessing automatic implicit associations between two target concepts and an attribute dimension on the basis of participants' reaction times and errors. In the IAT, participants have to sort stimuli from a computer screen into four different categories: (a) two contrasted target concept categories that form the target dimension and (b) two contrasted attribute categories that form the attribute dimension.

The IAT procedure

In the self-esteem IAT (Greenwald & Farnham, 2000; Rudolph, Schröder, & Schütz, 2006), as used in the example data, the target discrimination includes self-relevant (i.e., I, self, me, my, own, mine) and non-self-relevant (i.e., they, them, yours, you, others, their) words, and the attribute discrimination includes pleasant (i.e., holiday, joy, happiness, health, smile, and peace) and unpleasant (i.e., disaster, war, smell, grief, illness, and agony) words. The IAT consists of seven blocks altogether (with several trials per block), of which Blocks 1, 2, and 5 are the single or practice blocks, which introduce the target or attribute discrimination. In these blocks, the categories of either the target concepts or the attribute concepts are presented in the upper corners of each side (i.e., left and right) of the display screen. Participants are instructed to respond to exemplars of each category by pressing a key on the same side as the label. Blocks 3 and 4 as well as 6 and 7 are the combined blocks in which the attribute discrimination is paired with the target discrimination (i.e., participants must assign words from all four categories in these blocks). Thus, in the self-esteem IAT, in Blocks 3 and 4 (i.e., the compatible IAT phase), participants must respond to self-relevant and positive words with one key and to non-self-relevant and negative words with the other key. In Blocks 6 and 7 (i.e., the incompatible IAT phase), participants must respond to negative words and self-relevant words with one key and to positive words and non-self-relevant words
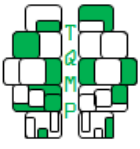
Table 1 ■ Computations of the four C measures and of the six different D measure

Details of the computation	C measures				D measures					
	C_1	C_2	C_3	C_4	D_1	D_2	D_3	D_4	D_5	D_6
Which blocks are included?	Include only test blocks (i.e., Blocks 4 and 7)			Include all blocks from the compatible and incompatible IAT phases (i.e., Blocks 3, 4, 6, and 7)						
Which trials are included?	Drop the first two trials of each block		Include all trials from Blocks 3, 4, 6, and 7							
Is a treatment applied to the upper tail?	Yes (Recode latencies > 3,000 ms to 3,000 ms)				Yes (Delete if latency > 10,000 ms)					
Is a treatment applied to the lower tail?	Yes (Recode latencies < 300 ms to 300 ms)				No	Yes (Delete if latency < 400 ms)	No		Yes (Delete if latency < 400 ms)	
How are errors treated?	Include all error latencies in analyses						Replace errors: mean(C) + 2 SD	Replace errors: mean(C) + 600 ms	Replace errors: mean(C) + 2 SD	Replace errors: mean(C) + 600 ms
Computational formula	$\mathrm{RT}_{\text{incompatible IAT phase}} - \mathrm{RT}_{\text{compatible IAT phase}}$				$\dfrac{\mathrm{RT}_{\text{incompatible IAT phase}} - \mathrm{RT}_{\text{compatible IAT phase}}}{SD \text{ in RT}}$					

Note. RT = reaction time, SD = standard deviation; mean(C) = mean correct latencies.

with the other key.[1]

The computation of IAT effects

The rationale behind the IAT is that the sorting task should be easier and thus completed more quickly when the two concepts sharing one response key are strongly associated. If they are weakly associated, sorting them into one category should be more difficult and should therefore be conducted more slowly. Greenwald, Nosek, and Banaji (2003a, 2003b) suggested a total of 10 algorithms (six D measures: D_1, D_2, D_3, D_4, D_5, D_6 and four conventional measures [i.e., C measures]: C_1, C_2, C_3, C_4) for computing the IAT effect. These algorithms indicate the strengths of the associations between the concepts. Greenwald et al. (2003a, 2003b) have extensively demonstrated the superiority of the measurement properties of the D measures over the C measures. Thus, they recommend that the D measures be used when computing IAT effects. This recommendation has also been supported by more recent research (e.g., Glashouwer, Smulders, de Jong, Roefs, & Wiers, 2013). Therefore, you might prefer to use D measures to compute the IAT effects for your own research. However, there may be circumstances where you need the C measures (e.g., If you want to compare your results with results from other studies that used the C measures). Therefore, we explain the mathematical details of all the traditional IAT effects and also provide R code for all of them.

The C measures. The four C measures may seem to be identical, but besides having some commonalities, they also have some differences. Let us take a look at this (see Table 1; Greenwald, Nosek, & Banaji, 2003a, 2003b). What the C measures have in common is that the latencies are recoded to deal with the lower and upper tails (i.e., reaction times < 300 ms are recoded to 300 ms and reaction times $> 3,000$ ms are recoded to 3,000 ms). In addition, all of the four C measures include all the reaction times (i.e., error latencies are not excluded). However, they differ in that they either include all trials from all blocks of the compatible and incompatible IAT phases (i.e., C_3 and C_4) or they exclude the first two trials of each block and only use the test blocks of each IAT phase (i.e., C_1 and C_2) for computation. Furthermore, they differ in their latency transformation. Prior to the calculation of the respective measure, the latencies are either transformed (i.e., a logarithmic transformation of the latencies occurs for C_1 and C_3) or not (i.e., C_2 and C_4).

The D measures. The D measures, too, seem more or less identical at first glance, but even though they have some commonalities and their computations look similar,

they also have some very specific differences (see Table 1; Greenwald et al., 2003a, 2003b). All the D measures represent a computed difference in reaction times between the blocks of the incompatible and compatible IAT phases, and this difference is adjusted for individual reaction times by dividing the difference by the individual standard deviation. Furthermore, all D measures include all of the trials from the compatible and incompatible IAT phases in their computation. Also, all latencies $> 10,000$ ms are deleted from them (i.e., the treatment of the RT in the upper tail). However, they differ in how the reaction times in the lower tail are treated. In some, the latencies < 400 ms are deleted (i.e., D_2, D_5, D_6), whereas in others, they are not (i.e., D_1, D_3, D_4). There are also differences in the treatment of errors. Two D measures use unaltered error latencies (i.e., D_1, D_2). For the other four, either the errors are replaced with the mean plus 2 SD of the correct latencies in the block in which the error occurred (i.e., D_3, D_5) or the errors are replaced by the block means of the correct latencies plus 600 ms (i.e., D_4, D_6).

The differences in the computations of IAT effects are reflected in research that has indicated that results depend on which IAT effect was used (e.g., Glashouwer et al., 2013; Röhner, 2017). Because researchers use different scoring algorithms to compute the IAT effect (e.g., Wilson & Scior, 2014), you should be able to understand what is included in the different IAT effects and how you can compute them in order to compare your results with those of other researchers. The present tutorial provides a detailed description of how to compute the traditional IAT effects (Greenwald et al., 2003a, 2003b) with the free software environment for statistical computing and graphics R (see also core team, 2019).

Step-by-step tutorial

Before you begin: Some prerequisites

Please ensure that you have installed R on your computer including the packages: "tidyverse", "data.table", and "reshape". In addition, you should prepare the data as described in Step 1.

Step 1: Data preparation

To use our R code, you should prepare your data as shown in Table 2. That is, you need a total of seven columns for your variables. Ensure that all of your columns are of the type numeric. The columns have to be arranged as follows: id (i.e., the subject ids of your participants, consisting of numbers only), order (i.e., whether the participants

---

[1]The presentation of the combined phases can be counterbalanced in IATs. Hence, as a researcher, one can decide whether the participant will work on the compatible phase first and afterwards on the incompatible one (i.e., the sequence of the IAT presented above) or whether the phases should be presented the other way around. In order to avoid unnecessarily complicating the description of the IAT, only the most common order is presented in this paper.

Table 2 ■ First ten rows of the example data (i.e., "iat_data.dat")

id	order	test	blocknum	trialnum	correct	latency
1	1	1	3	1	1	812
1	1	1	3	2	1	687
1	1	1	3	3	1	723
1	1	1	3	4	1	619
1	1	1	3	5	1	657
1	1	1	3	6	1	580
1	1	1	3	7	1	596
1	1	1	3	8	1	687
1	1	1	3	9	1	618
1	1	1	3	10	1	936

Note. id (i.e., the subject ids of your participants that consists of numbers only), order (i.e., whether the participants started with the compatible or incompatible IAT phase; 1 = start with compatible phase followed by the incompatible phase; 2 = start with the incompatible phase followed by the compatible phase), test (i.e., whether the block represents a practice or a test block; 1 = practice block; 2 = test block), blocknum (i.e., represents the number of the IAT block), trialnum (i.e., represents the number of the respective trial within the IAT blocks), correct (i.e., whether the answers of the participants on the trials were correct [1] or incorrect [0]), and latency (i.e., the latencies in milliseconds of participants when answering to the trials).

began with the compatible or incompatible IAT phase; 1 = began with compatible phase followed by the incompatible phase; 2 = began with the incompatible phase followed by the compatible phase), test (i.e., whether the block represents a practice or a test block; 1 = practice block; 2 = test block), blocknum (i.e., the number of the IAT block), trialnum (i.e., the number of the respective trial within an IAT block), correct (i.e., whether the participants' answers on the trials were correct [1] or incorrect [0]), and latency (i.e., the number of milliseconds it took participants to give their answers in the trials). Rename your data file: "iat_data.dat". Save your data file in the "data" folder.

Step 2: Decide which IAT effect(s) you need to compute

Based on your study and on your study's questions, consider which IAT effects need to be computed. For example, if you want to replicate a result from a specific study that used a certain IAT effect (e.g., the D_2 measure), you should also use the same IAT effect that was used in the original study. After you make your decision, please select and open the respective R script (i.e., for the conventional measures: "C_1.R" to compute C_1, "C_2.R" to compute C_2, "C_3.R" to compute C_3, and "C_4.R" to compute C_4; for the D measures: "D_1.R" to compute D_1, "D_2.R" to compute D_2, "D_3.R" to compute D_3, "D_4.R" to compute D_4, "D_5.R" to compute D_5, and "D_6.R" to compute D_6). If you need all IAT effects, please select and open the R script "all_measures.R."

Step 3: Compute your IAT effect(s)

As we explained in the introduction, the traditional IAT effects have similarities and differences in their computation. Therefore, some computation scripts are more similar to each other than others (e.g., the computations and thus also the scripts for the C measures are more similar to each other than is true for the D measures). In order to avoid unnecessarily complicating and lengthening the description, we provide complete descriptions for how to compute the C_1 and D_1 representatives, and then we note the relevant differences in the other scripts afterwards.

How to compute the conventional IAT effects $C_1 – C_4$. Using C_1 as an example, you should see the script shown in Listing 1 after you open "C_1.R." The script does the following for you:

Part 1. loads the library "tidyverse,"

Part 2. loads the raw data (i.e., the "iat_data.dat" that you saved in the "data" folder),

Part 3. preprocesses these raw data (i.e., drops the first two trials of each block according to the conventional procedure, recodes the latencies of the trials that are under 300 or greater than 3,000 ms [latencies < 300 ms will be recoded to 300 ms and latencies $> 3,000$ ms will be recoded to 3,000 ms], calculates the natural logarithm of the respective latencies, and calculates the mean of the log latencies for each of the IAT blocks that were used),

Part 4. prepares the calculation (i.e., filters the blocks so that only test blocks are used for the calculation and rearranges the data to prepare them for the calculation),

Part 5. calculates the score (i.e., the actual calculation

step), and

Part 6. writes an output for it.

To compute C_1, you simply have to run the whole script. Afterwards, you can find the output (i.e., "C_1-YYYY-MM-DD.dat") in the "output" folder.

For C_2, C_3, and C_4, you can also simply run the respective R scripts. The differences in the scripts concern Part 3 (the preprocessing of the raw data), Part 4 (the preparation of the calculation), and Part 5 (the calculation of the score). Different from the C_1 script, the C_2 script does not transform the latencies into log latencies in Part 3 (the preprocessing of the raw data).

Different from the C_1 script, the C_3 script does not drop the first two trials of each IAT block in Part 3 (the preprocessing of the raw data), selects practice and test blocks for calculation in Part 4 (the preparation of the calculation), and uses them in Part 5 (the calculation of the score).

Different from the C_1 script, the C_4 script does not drop the first two trials of each IAT block and does not transform the latencies to log latencies in Part 3 (the preprocessing of the raw data), selects practice and test blocks for calculation in Part 4 (the preparation of the calculation), and uses them in Part 5 (the calculation of the score).

How to compute the D measure IAT effects D_1 – D_6. Using D_1 as an example, you should see the script shown in Listing 2 after opening "D_1.R." The script does the following for you:

Part 1. loads the library "tidyverse,"

Part 2. loads the raw data (i.e., the "iat_data.dat" that you saved in the "data" folder),

Part 3. preprocesses these raw data (i.e., drops trials with latencies $> 10,000$ ms and calculates the mean and standard deviation of the latencies for each participant),

Part 4. prepares the calculation (i.e., calculates the mean of the latencies for each IAT block and for each person, selects both the practice and test blocks for the calculation; rearranges the data to prepare them for the calculation),

Part 5. calculates the score (i.e., the actual calculation step), and

Part 6. writes an output for it.

To compute D_1, you simply have to run the whole script. Afterwards, you can find the output (i.e., "D_1-YYYY-MM-DD.dat") in the "output" folder.

For D_2, D_3, D_4, D_5, and D_6, you can also simply run the respective scripts. The differences in the scripts concern Part 3 (the preprocessing of the raw data), Part 4 (the preparation of the calculation), and Part 5 (the calculation of the score). Different from the D_1 script, the D_2 script additionally drops trials with latencies < 400 ms in Part 3 (the preprocessing of the raw data), selects the practice and test blocks for calculation in Part 4 (the preparation of

the calculation), and uses them Part 5 (the calculation of the score).

Different from the D_1 script, the D_3 script calculates the mean and standard deviation of the correct latencies (i.e., latencies stemming from trials that have been answered correctly by the participant) for each block for each participant in Part 3 (the preprocessing of the raw data), and additionally corrects the error latencies (i.e., latencies stemming from trials that have been incorrectly answered by participants) by replacing them (i.e., with the sum of the mean latencies of the correct responses per block and the standard deviation of the correct responses per block multiplied by 2), calculates the penalty (i.e., the mean latencies of the corrected latencies per block and participant), selects the practice and test blocks for the calculation in Part 4 (the preparation of the calculation), and uses them in the calculation.

Different from the D_1 script, the D_4 script calculates the mean and the standard deviation of the correct latencies (i.e., latencies stemming from trials that have been answered correctly by the participant) for each participant in Part 3 (the preprocessing of the raw data) and additionally corrects the error latencies (i.e., latencies stemming from trials that have been incorrectly answered by participants) by the sum of the mean latency of the correct responses plus 600 ms, calculates the penalty (i.e., the mean latencies of the corrected latencies per block and participant), selects the practice and test blocks for calculation in Part 4 (the preparation of the calculation), and uses them in the calculation.

Different from the D_1 script, the D_5 script drops trials with latencies < 400 ms, calculates the mean and standard deviation of the correct latencies (i.e., latencies stemming from trials that have been answered correctly by the participant) for each block for each participant in Part 3 (the preprocessing of the raw data), and additionally corrects the error latencies (i.e., latencies stemming from trials that have been incorrectly answered by participants) by replacing them (i.e., with the sum of the mean latencies of the correct responses per block and the standard deviation of the correct responses per block multiplied by 2), calculates the penalty (i.e., the mean latencies of the corrected latencies per block and participant), selects the practice and test blocks for calculation in Part 4 (the preparation of the calculation), and uses them in the calculation.

Different from the D_1 script, the D_6 script drops trials with latencies < 400 ms, calculates the mean and standard deviation of the correct latencies (i.e., latencies stemming from trials that have been answered correctly by the participant) for each participant in Part 3 (the preprocessing of the raw data), and additionally corrects the error latencies (i.e., latencies stemming from trials that have been incor-

Figure 1 ■ Graphical output 1 of "quality_checks.R" (i.e., "quality-checks-col-2019-06-19.pdf").
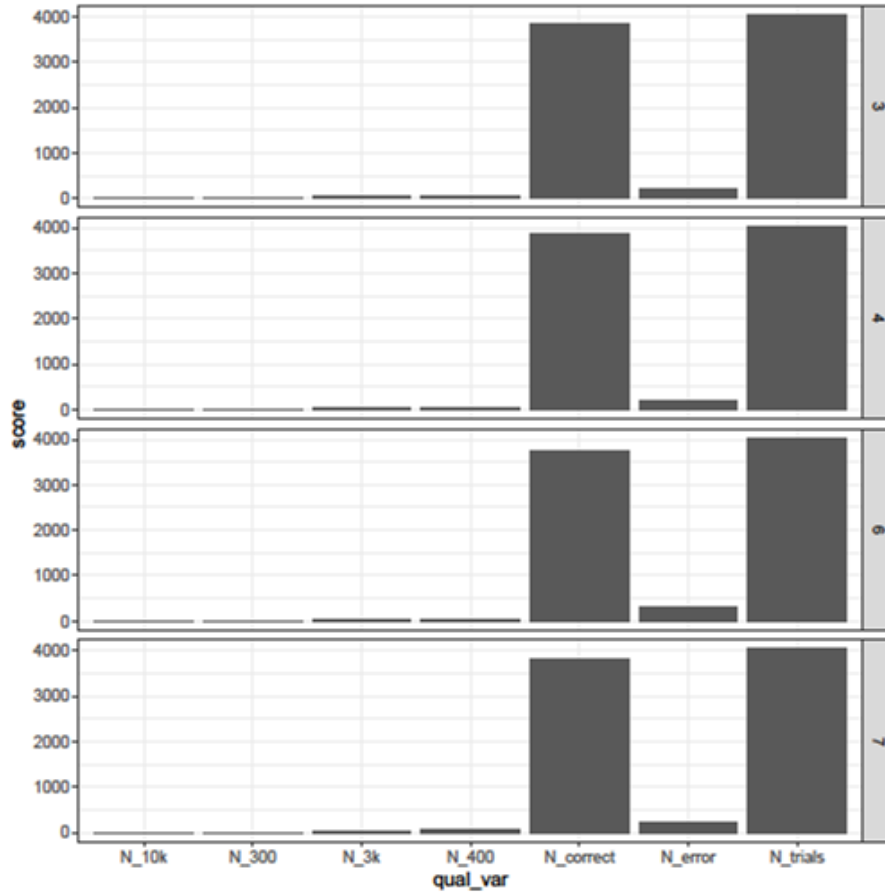
rectly answered by participants) by the sum of the mean latency of the correct responses plus 600 ms, calculates the penalty (i.e., the mean latencies of the corrected latencies per block and participant), selects the practice and test blocks for calculation in Part 4 (the preparation of the calculation), and uses them in the calculation.

How to compute all the traditional IAT effects. If you want to compute all of the traditional IAT effects, please select the script "all_measures.R" (see Listing 3) and run it. The script does the following for you:

1. loads the libraries "data.table" and "reshape,"

2. sources all 10 scripts we described above and generates the respective output files,

3. reads and combines them,

4. and generates a combined output file.

You can find this output (i.e., "all-measures-YYYY-MM-DD.dat") in the "output" folder. Make sure that you have emptied the output folder before running this script because otherwise, the data will be combined incorrectly.

Step 4: Check the IAT effects and use them for further calculation

You may want to check the quality of your IAT effects. Therefore, select the script "quality_checks.R" and run it. The script does the following for you:

1. loads the library "tidyverse,"

2. loads the raw data,

3. summarizes the data, calculates the quality criteria (i.e., mean latency, standard deviation of the latencies, mean of correct responses, standard deviation of correct responses, total number of trials, number of correct trials, number of error trials, number of trials < 300 ms, number of trials $> 3,000$ ms, number of trials < 400 ms, number of trials $> 10,000$ ms, percentage of error trials, percentage of trials < 300 ms, percentage of trials $> 3,000$ ms, percentage of trials < 400 ms, and percentage of trials $> 10,000$ ms per person and per IAT block), and provides a data frame with the minimum and maximum values of
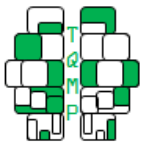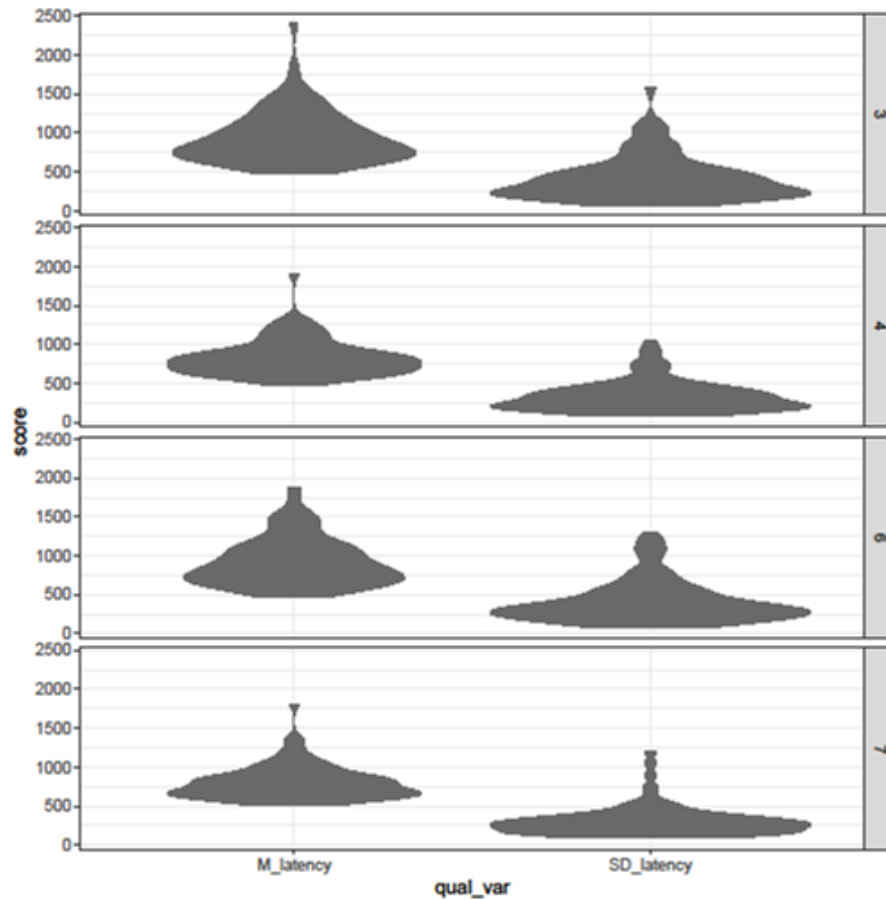
Figure 2 ■ Graphical output 2 of "quality_checks.R" (i.e., "quality-checks-vio-2019-06-19.pdf").

the respective quality criteria,

4. creates bar plots to visually represent the total number of trials, number of correct trials, number of error trials, number of trials < 300 ms, number of trials $> 3,000$ ms, number of trials < 400 ms, number of trials $> 10,000$ ms per IAT block as well as violin plots to visually represent the reaction time distribution (i.e., mean and standard deviation per IAT block), and

5. writes the output files for the resulting quality criteria and for the plots.

After checking the quality of the IAT effects, you can use them for further analyses.

Results

Following the protocol above will provide you with the traditional IAT effects that you have selected. To convey an impression of what your results should look like, we stored the outputs of example data that were used in a previous study (Röhner, Schröder-Abé, & Schütz, 2011) at OSF (https://osf.io/abrz6/) and on this journal's web site. The

name of the files was constructed according to the principle: "name of the measure-date of the computation (i.e., YYYY-MM-DD).dat" in each case (i.e., "C1-2019-06-19.dat" shows the output for C_1, "C2-2019-06-19.dat" shows the output for C_2, and so on). Using C_1 as an example, you should see the following after opening "C1-2019-06-19.dat": Column 1 shows the respective ID, and column 2 shows the respective IAT effect (here C_1).

In addition, you will obtain data on the quality of your IAT effects. You will obtain the document "quality-checks-YYYY-MM-DD.dat," "quality-checks-col-YYYY-MM-DD.pdf," and "quality-checks-vio-YYYY-MM-DD.pdf" (see Table 3; Figures 1 and 2). When you open our example data (i.e., "quality-checks-2019-06-19.dat," "quality-checks-col-2019-06-19.pdf," and "quality-checks-vio-2019-06-19.pdf"), you may see, for example, that the number of correct trials in our example data ranges from 26 to 48, the number of trials faster than 300 ms ranges from 0 to 2, the number of trials slower than 3,000 ms ranges from 0 to 11, the number
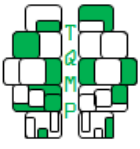
Table 3 ∎ First ten rows of the quality checks output data (i.e., "quality-checks-YYYY-MM-DD.dat")

id	blocknum	M_latency	SD_latency	M_correct	SD_correct	N_trials	N_correct	N_error	N_300	N_3k	N_400	N_10k	P_error	P_300	P_3k	P_400	P_10k
1	3	628	135.892	1	0	48	48	0	0	0	1	0	0	0	0	0.021	0
1	4	614	207.401	0.979	0.144	48	47	1	0	0	0	0	0.021	0	0	0	0
1	6	1018	406.189	0.812	0.394	48	39	9	0	0	0	0	0.188	0	0	0	0
1	7	837	325.241	0.938	0.245	48	45	3	0	0	0	0	0.062	0	0	0	0
2	3	737	245.248	0.854	0.357	48	41	7	0	0	1	0	0.146	0	0	0.021	0
2	4	673	252.451	0.896	0.309	48	43	5	0	0	0	0	0.104	0	0	0	0
2	6	981	370.032	0.875	0.334	48	42	6	0	0	0	0	0.125	0	0	0	0
2	7	842	329.07	0.917	0.279	48	44	4	0	0	0	0	0.083	0	0	0	0
3	3	527	111.689	0.938	0.245	48	45	3	0	0	3	0	0.062	0	0	0.062	0
3	4	590	202.176	0.875	0.334	48	42	6	0	0	1	0	0.125	0	0	0.021	0

Note. id (i.e., the subject ids of your participants that consists of numbers only), blocknum (i.e., number of the IAT block), Quality criteria per id and blocknum: M_latency (i.e., mean latency of all trials without treatment), SD_latency (i.e., standard deviation of latencies of all trials without treatment), M_correct (i.e., mean latency of all correct trials without treatment), SD_correct (i.e., standard deviation of latencies of all correct trials without treatment), N_trials (i.e., total number of trials), N_correct (i.e., number of correct trials), N_error (i.e., number of error trials), N_300 (i.e., number of trials with latencies < 300 ms), N_3k (i.e., number of trials with latencies > 3,000 ms), N_400 (i.e., number of trials with latencies < 400 ms), N_10k (i.e., number of trials with latencies > 10,000 ms), P_error (i.e., percentage of error trials), P_300 (i.e., percentage of trials with latencies < 300 ms), P_3k (i.e., percentage of trials with latencies > 3,000 ms), P_400 (i.e., percentage of trials with latencies < 400 ms), P_10k (i.e., percentage of trials with latencies > 10,000 ms).

of trials faster than 400 ms ranges from 0 to 14, and there are no trials slower than 10,000 ms.

If you take a look at our example data "quality-checks-col-2019-06-19.pdf," you will see from right to left that the number of all trials (i.e., "N_trials") is the same in all IAT blocks (i.e., as expected when all IAT blocks have the same length). This might serve as a check for whether the whole IAT ran correctly. Also, you will see that the number of incorrect trials (i.e., "N_error") is much lower than the number of correct ones (i.e., "N_correct") across all IAT blocks (i.e., as expected when participants did not randomly press keys). This might also serve as a check for whether participants actually followed your instructions. Last but not least, you will see that there are only a few trials that are recoded or filtered by the IAT algorithms because the number of trials < 400 ms (i.e., "N_400"), $> 3,000$ ms (i.e., "N_3k"), < 300 ms (i.e., "N_300"), and $> 10,000$ ms (i.e., "N_10k") is low and also spread out equally across the IAT blocks (i.e., as expected when participants follow the standard IAT instructions). This might serve as a check for whether the participants followed the instructions to react as quickly as possible while simultaneously trying not to make mistakes.

If you take a look at "quality-checks-vio-2019-06-19.pdf," you will see the distribution of the mean latencies and the standard deviation of the latencies across the IAT blocks for our example data. You can see that the majority of participants have a mean reaction time ranging from 466 ms to 1,500 ms across the IAT blocks (i.e., as expected, participants usually did not exceed the maximum reaction time of 10,000 ms or fall under the minimum of 300 ms). Again, this is a hint that participants worked on the IAT as instructed and were not distracted (e.g., They did not watch videos of cats on the Internet while participating in your study). Also, you can see that the majority of the participants have a standard deviation ranging from 60 to 750 across the IAT blocks. Huge standard deviations would be an indicator that something went wrong because they show that there were large differences in participants' response times.

Discussion

In this tutorial, we present R code and example data to compute the 10 traditional IAT effects (Greenwald et al., 2003a, 2003b). We want to end our tutorial by answering some questions that might have arisen for readers after absorbing all the information we provided above.

Which IAT effect should I compute?

Greenwald et al. (2003a, 2003b) have extensively demonstrated the superiority of the measurement properties of the D measures over the conventional measures. Accord-

ingly, they recommend that D measures be used when computing IAT effects. This recommendation has also been backed up by more recent research (e.g., Glashouwer et al., 2013). Therefore, in most cases, the D measures will be the way to go when analyzing IAT effects. However, there may be specific circumstances where you need the conventional measures, too. It depends on your study's conditions and the hypotheses you are testing in your research. Therefore, it is difficult to provide recommendations for every possible situation. Instead, we want to give you some key issues that may play a role when you choose "your" IAT effects.

For example, you may want to compare your results with those of another study. In this case, it will be useful to use the same IAT effect as the authors of the study that you want to compare yours to. The same is true for any kind of replication studies. In turn, if you want to meta-analyze data and the authors of the studies you want to include used different IAT effects, then it would be best to choose one IAT effect and re-calculate the rest accordingly.

Are there alternatives to computing IAT effects?

It is important to keep in mind that the traditional IAT effects do not decompose the IAT's results, for example, in terms of association-related versus non-association-related processes. Recent research has highlighted the idea that diffusion model analyses (Klauer, Voss, Schmitz, & Teige-Mocigemba, 2007; Röhner & Ewers, 2016b) or the quad model (Conrey, Sherman, Gawronski, Hugenberg, & Groom, 2005) are useful for decomposing the processes underlying the IAT effect in a more detailed manner. If you want to add diffusion-model-based IAT effects to the traditional IAT effects, the tutorial by Röhner and Ewers (2016a) as well as the paper by Röhner and Thoss (2018) can provide additional information and a guide for how to do this.

Summary and conclusion

This step-by-step tutorial along with the R scripts should enable readers to easily compute the 10 traditional IAT effects with the open source tool R. Example data are stored at OSF (https://osf.io/abrz6/) and this journal's web site in order to facilitate understanding of how the data should be prepared and what the output will look like. Also, we discussed ways to figure out which IAT effect should be computed and possible alternatives to the computation of traditional IAT effects.

Authors' note

Correspondence concerning this article should be addressed to Jessica Röhner, Department of Psychology, University of Bamberg, Markusplatz 3, 96047 Bamberg, Germany.

References

Bosson, J. K., Swann, W. B., Jr., & Pennebaker, J. W. (2000). Stalking the perfect measure of implicit self-esteem: The blind men and the elephant revisited? *Journal of Personality and Social Psychology*, *79*, 631–643. doi:10.1037/0022-3514.79.4.631

Conrey, F. R., Sherman, J. W., Gawronski, B., Hugenberg, K., & Groom, C. J. (2005). Separating multiple processes in implicit social cognition: The quad model of implicit task performance. *Journal of Personality and Social Psychology*, *89*, 469–487. doi:10.1037/0022-3514.89.4.469

core team, R. (2019). R: The r project for statistical computing. [online] available at: [accessed 14 jun. 2019]. Retrieved from https://www.r-project.org

Glashouwer, K. A., Smulders, F. T. Y., de Jong, P. J., Roefs, A., & Wiers, R. W. H. J. (2013). Measuring automatic associations: Validation of algorithms for the Implicit Association Test (IAT) in a laboratory setting. *Journal of Behavior Therapy and Experimental Psychiatry*, *44*, 105–113. doi:10.1016/j.jbtep.2012.07.015

Greenwald, A., & Farnham, S. (2000). Using the Implicit Association Test to measure self-esteem and self-concept. *Journal of Personality and Social Psychology*, *79*, 1022–1038. doi:10.1037/0022-3514.79.6.1022

Greenwald, A., McGhee, D., & Schwartz, J. (1998). Measuring individual differences in implicit cognition: The Implicit Association Test. *Journal of Personality and Social Psychology*, *74*, 1464–1480. doi:10.1037/0022-3514.74.6.1464

Greenwald, A., Nosek, B., & Banaji, M. (2003a). Understanding and using the Implicit Association Test: I. An improved scoring algorithm. *Journal of Personality and Social Psychology*, *85*, 197–216. doi:10.1037/0022-3514.85.2.197

Greenwald, A., Nosek, B., & Banaji, M. (2003b). Understanding and using the Implicit Association Test: I. An improved scoring algorithm: Correction to greenwald et al. (2003). *Journal of Personality and Social Psychology*, *85*, 481. doi:10.1037/h0087889

Klauer, K. C., Voss, A., Schmitz, F., & Teige-Mocigemba, S. (2007). Process components of the Implicit Associa-tion Test: A diffusion-model analysis. *Journal of Personality and Social Psychology*, *93*, 353–368. doi:10.1037/0022-3514.93.3.353

Röhner, J. (2017). *The impact of working memory load on nonfaked and faked IAT* D *measures*. Dresden.

Röhner, J., & Ewers, T. (2016a). How to analyze (faked) Implicit Association Test data by applying diffusion model analyses with the fast-dm software: A companion to Röhner & Ewers (2016). *The Quantitative Methods for Psychology*, *12*, 220–231. doi:10/20982/tqmp.12.3.p220

Röhner, J., & Ewers, T. (2016b). Trying to separate the wheat from the chaff: Construct- and faking-related variance on the Implicit Association Test (IAT). *Behavior Research Methods*, *48*, 243–258. doi:10.3758/s13428-015-0568-1

Röhner, J., Schröder-Abé, M., & Schütz, A. (2011). Exaggeration is harder than understatement, but practice makes perfect! Faking success in the IAT. *Experimental Psychology*, *58*, 464–472. doi:10.1027/1618-3169/a000114

Röhner, J., & Thoss, P. J. (2018). EZ: An easy way to conduct a more fine-grained analysis of faked and nonfaked implicit association test (IAT) data. *The Quantitative Methods for Psychology*, *14*, 17–35. doi:10.20982/tqmp.14.1.p017

Rudolph, A., Schröder-Abé, M., Schütz, A., Gregg, A. P., & Sedikides, C. (2008). Through a glass, less darkly? Reassessing convergent and discriminant validity in measures of implicit self-esteem. *European Journal of Psychological Assessment*, *24*, 273–281. doi:10.1027/1015-5759.24.4.273

Rudolph, A., Schröder, M., & Schütz, A. (2006). Ein Impliziter Assoziationstest zur Erfassung von Selbstwertschätzung [An Implicit Association Test for self-esteem]. In T. M. Ortner, R. T. Proyer, & K. D. Kubinger (Eds.), *Theorie und praxis objektiver persönlichkeitstests* (pp. 153–163). Bern: Huber.

Wilson, M. C., & Scior, K. (2014). Attitudes towards individuals with disabilities as measured by the Implicit Association Test: A literature review. *Research in Developmental Disabilities*, *35*, 294–321. doi:10.1016/j.ridd.2013.11.003

Appendix A: Listings

Listing 1: Example source code of the C_1 measure (i.e., "C1.R").

```
# libraries ----
library(tidyverse)

# load raw data ----
iat_raw <- read.delim("data/iat_data.dat",
```

```r
  header = TRUE,
  sep = "\t",
  fill = TRUE,
  colClasses = rep("numeric", 7)
)

# preprocess raw data ----
iat_1 <- iat_raw %>%
  filter(
    # drop the first two trials of each block according to the conventional procedure
    trialnum > 2
  ) %>%
  mutate(
    # recode latencies
    latency = if_else(latency < 300, 300,
      if_else(latency > 3000, 3000, latency)
    ),
    # natural logarithm of respective latency
    ln_lat = log(latency)
  ) %>%
  # calculate mean for natural logarithm of latencies
  group_by(id, order, test, blocknum) %>%
  summarise(
    M_ln_lat = mean(ln_lat)
  ) %>%
  ungroup()

# prepare calculation ----
dd <- iat_1 %>% select(id, order) %>% unique()

for (i in c(4, 7)) {
  iat_b <- iat_1 %>%
    filter(blocknum == i) %>%
    select(-c(test, blocknum))
  names(iat_b) <- c("id", "order", paste0("M_ln_lat_", i))
  dd <- full_join(dd, iat_b, by = c("id", "order"))
}

# calculate score ----
iat_out <- dd %>%
  mutate(
    # C1 (conventional logarithmic test)
    C1 = round(case_when(
      order == 1 ~ (M_ln_lat_7 - M_ln_lat_4),
      order == 2 ~ (M_ln_lat_4 - M_ln_lat_7)
    ), 2)
  ) %>%
  select(id,C1)

# write output to file ----
write.table(iat_out,
  paste0("output/C1-", Sys.Date(), ".dat"), sep = "\t", quote = FALSE,
```

```
    row.names = FALSE
)
```

Listing 2: Example source code of the D_1 measure (i.e., "D1.R").

```r
# libraries  ----
library(tidyverse)

# load raw data ----
iat_raw <- read.delim("data/iat_data.dat",
  header = TRUE,
  sep = "\t",
  fill = TRUE,
  colClasses = rep("numeric", 7)
)

# preprocess raw data ----
iat_1 <- iat_raw %>%
  filter(
    # filter  latencies
    latency <= 10000
  )

# prepare D-asis
iat_2 <- iat_1 %>%
  group_by(
    id, order, test
  ) %>%
  summarise(
    N_x = n(),
    M_X_lat = mean(latency),
    SD_X_lat = sd(latency)
  ) %>%
  ungroup()

# prepare calculation  ----
iat_3 <- full_join(iat_1, iat_2) %>%
  mutate(D_asis = latency) %>%
  group_by(id, order, test, blocknum, SD_X_lat) %>%
  summarise(
    N_3 = n(),
    M_D_asis = mean(D_asis)
  ) %>%
  ungroup() %>%
  arrange()

dd <- iat_3 %>% select(id, order) %>% unique()

for (i in c(3, 4, 6, 7)) {
  iat_b <- iat_3 %>%
    filter(blocknum == i) %>%
    select(-c(test, blocknum))
```

```
  names(iat_b) <- c("id", "order", paste0("SD_X_lat_", i), paste0("N_3_", i),
    paste0("M_D_asis_", i))
  dd <- full_join(dd, iat_b, by = c("id", "order"))
}

# calculate D-scores ----
iat_out <- dd %>%
  mutate(
    # D-asis practise
    D_asis_p = case_when(
      order == 1 ~ (M_D_asis_6 - M_D_asis_3) / SD_X_lat_3,
      order == 2 ~ (M_D_asis_3 - M_D_asis_6) / SD_X_lat_3
    ),
    # D-asis test
    D_asis_t = case_when(
      order == 1 ~ (M_D_asis_7 - M_D_asis_4) / SD_X_lat_4,
      order == 2 ~ (M_D_asis_4 - M_D_asis_7) / SD_X_lat_4
    ),
    # D-asis practise and test
    D_asis = round((D_asis_p + D_asis_t) / 2, 3),
    # N trials practise
    N_p = N_3_3 + N_3_6,
    # N trials test
    N_t = N_3_4 + N_3_7
  ) %>%
  select(id, D1 = D_asis)

# write output to file ----
write.table(iat_out,
    paste0("output/D1-", Sys.Date(), ".dat"), sep = "\t", quote = FALSE,
    row.names = FALSE
)
```

Listing 3: Example source code to calculate all traditional IAT-effects at once (i.e., "all_measures.R"). The code calls and runs the scripts for C_1–C_4 and D_1–D_6 and combines the respective outputs to a single file.

```
# libraries ----
library("data.table")
library("reshape")

# source all ----
for (i in c(
  "C1", "C2", "C3", "C4", # conventional IAT measures
  "D1", "D2", "D3", "D4", "D5", "D6" # D-measures
)) {
  source(paste0(i, ".R"))
}

# combine all output files from sourced scripts ----
temp_wd <- getwd()
setwd("output")
```
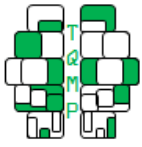
```
all_measures <- as.data.frame(
    merge_recurse(lapply(list.files(), fread, sep = "\t"))
)

# write an output file  ————
write.table(all_measures, paste0("all-measures-", Sys.Date(), ".dat"),
    sep = "\t", quote = FALSE, row.names = FALSE
)

setwd(temp_wd)
```

Open practices

🔵 The *Open Data* badge was earned because the data of the experiment(s) are available on the journal's web site.
🟠 The *Open Material* badge was earned because supplementary material(s) are available on the journal's web site.

Citation

Röhner, J., & Thoss, P. J. (2019). A tutorial on how to compute traditional iat effects with R. *The Quantitative Methods for Psychology*, *15*(2), 134–147. doi:10.20982/tqmp.15.2.p134